Fifty-fourth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 27 - 30, 2016

Gambler's Ruin Bandit Problem

Nima Akbarzadeh, Cem Tekin
Bilkent University, Electrical and Electronics Engineering Department, Ankara, Turkey

***Abstract*—In this paper, we propose a new multi-armed bandit problem called the *Gambler's Ruin Bandit Problem* (GRBP). In the GRBP, the learner proceeds in a sequence of rounds, where each round is a Markov Decision Process (MDP) with two actions (arms): a *continuation action* that moves the learner randomly over the state space around the current state; and a *terminal action* that moves the learner directly into one of the two terminal states (goal and dead-end state). The current round ends when a terminal state is reached, and the learner incurs a positive reward only when the goal state is reached. The objective of the learner is to maximize its long-term reward (expected number of times the goal state is reached), without having any prior knowledge on the state transition probabilities. We first prove a result on the form of the optimal policy for the GRBP. Then, we define the regret of the learner with respect to an *omnipotent oracle*, which acts optimally in each round, and prove that it increases logarithmically over rounds. We also identify a condition under which the learner's regret is bounded. A potential application of the GRBP is optimal medical treatment assignment, in which the continuation action corresponds to a conservative treatment and the terminal action corresponds to a risky treatment such as surgery.**

I. Introduction

Multi-armed bandits (MAB) are used to model a plethora of applications that require sequential decision making under uncertainty ranging from clinical trials [1] to web advertising [2]. In the conventional MAB [3], [4] the learner chooses an action from a finite set of actions at each round, and receives a random reward. The goal of the learner is to maximize its long-term expected reward by choosing actions that yield high rewards. This is a non-trivial task, since the reward distributions are not known beforehand. Numerous order-optimal index-based learning rules have been developed for the conventional MAB [4]–[6]. These rules act myopically by choosing the action with the maximum index in each round.

Situations that require multiple actions to be taken in each round cannot be modeled using conventional MAB. As an example, consider medical treatment administration. At the beginning of each round a patient arrives to the *intensive care unit* (ICU) with a random initial health state. The goal state is defined as *discharge* and dead-end state is defined as *death*. Actions correspond to treatment options that move the patient randomly over the state space. The objective is to maximize the expected number of patients that are discharged by learning the optimal treatment policy using the observations gathered from the previous patients. In the example given above, each round corresponds to a goal-oriented Markov Decision Process (MDP) with dead-ends [7]. The learner knows the state space, goal and dead-end states, but does not know the state transition probabilities a priori. At each round, the learner chooses a sequence of actions and only observes the state transitions that result from the chosen actions. In the literature, this kind of feedback information is called *bandit feedback* [8].

Motivated by the application described above, we propose a new MAB problem in which multiple arms are selected in each round until a terminal state is reached. Due to its resemblance to the *Gambler's Ruin Problem* [9]–[11], we call this new MAB problem the *Gambler's Ruin Bandit Problem* (GRBP). In GRBP, the system proceeds in a sequence of rounds $\rho \in \{1, 2, \ldots\}$. Each round is modeled as an MDP (as in Fig. 1) with unknown state transition probabilities and terminal (absorbing) states. The set of terminal states includes a *goal state* G and a *dead-end state* D, and the non-terminal states are ordered between the goal and dead-end states. In each non-terminal state, there are two possible actions: a *continuation action* (action C) that moves the learner randomly over the state space around the current state; and a *terminal action* (action F) that moves the learner directly into a terminal state. Starting from a random, non-terminal initial state, the learner chooses a sequence of actions and observes the resulting state transitions until a terminal state is reached. The learner incurs a unit reward if the goal state is reached. Otherwise, it incurs no reward. The goal of the learner is to maximize its cumulative expected reward over the rounds.

If the state transition probabilities were known beforehand, an *omnipotent oracle* with unlimited computational power could calculate the optimal policy that maximizes the probability of hitting the goal state from any initial state, and then select its actions according to the optimal policy. We define the regret of the learner by round ρ as the difference in the expected number of times the goal state is reached by the omnipotent oracle and the learner by round ρ.

First, we show that the optimal policy for GRBP can be computed in a straightforward manner: there exists a threshold state above which it is always optimal to take action C and on or below which it is always optimal to take action F. Then, we propose an online learning algorithm for the learner, and bound its regret for two different regions that the actual state transition probabilities can lie in. The regret is bounded (finite) in one region, while it is logarithmic in the number of rounds in the other region. These bounds are problem-specific, in the sense that they are functions of the state transition probabilities. Finally, we illustrate the

Cem Tekin is supported by TUBITAK 2232 Fellowship (116C043).

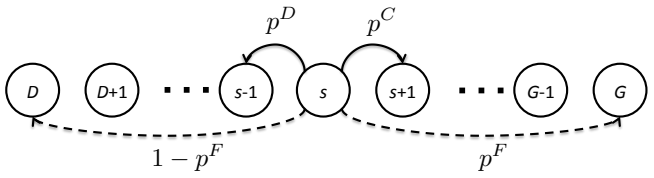

Figure 1. State transition model of the GRBP. Only state transitions out of state s are shown. Dashed arrows correspond to possible state transitions by taking action F, while solid arrows correspond to possible state transitions by taking action C. Weights on the arrows correspond to state transition probabilities. The state transition probabilities for all other non-terminal states are the same as state s.

behavior of the regret as a function of the state transition probabilities through numerical experiments.

The contributions of this paper can be summarized as follows:

- We define a new MAB problem, called GRBP, in which the learner takes a sequence of actions in each round with the objective of reaching to the goal state.
- We show that using conventional MAB algorithms such as UCB1 [4] in GRBP by enumerating all deterministic Markov policies is very inefficient and results in high regret.
- We prove that the optimal policy for GRBP has a threshold form and the value of the threshold can be calculated in a computationally efficient way.
- We derive bounds on the regret of the learner with respect to an omnipotent oracle that acts optimally. Unlike conventional MAB where the regret growth is at least logarithmic in the number of rounds [3], in GRBP regret can be either logarithmic or bounded, based on the values of the state transition probabilities. We explicitly characterize the region of state transition probabilities in which the regret is bounded.

Remainder of the paper is organized as follows. Related work is given in Section II. GRBP is defined in Section III. Form of the optimal policy for the GRBP is given in Section IV. The learning algorithm for GRBP is given in Section V together with its regret analysis. Numerical results are shown in Section VI. Conclusion is given in Section VII.

II. Related Work

A. Gambler's Ruin Problem

If action F is removed from the GRBP, it becomes the Gambler's Ruin Problem. In the model of Hunter et al. [10] of the Gambler's Ruin Problem, in addition to the standard outcome of moving one state to the left or right, two extra outcomes are also considered. One outcome changes the state immediately to G, while the other outcome changes the state immediately to D. These outcomes are referred to as *Windfall* and *Catastrophe* outcomes, respectively. The ruin and winning probabilities and the duration of the game are calculated based on these additional outcomes. In another model [11], modifications such as the chance of absorption in states other than G and D and staying in the same state are considered. The ruin and winning probabilities are calculated according to the proposed state transition model. Unlike GRBP which is an MDP, the Gambler's Ruin Problem is a Markov chain. Moreover, the ruin and winning probabilities in the models above can be calculated exactly since the transition probabilities are assumed to be known.

B. MDPs

GRBP is closely related to goal oriented MDPs and stochastic shortest path problems [12]. For these problems, in each state (or time epoch), an action has to be taken with the aim of reaching to the goal state (G) with minimum cost. For this task, the optimal policy have to be determined beforehand using the set of known transition probabilities. Recently, progress has been made in obtaining solutions for MDPs that have dead-end (D) states in addition to goal (G) states [7], [13]. These solutions require value iteration and heuristic search methods to be performed using the knowledge of transition probabilities. To the best of our knowledge, a reinforcement learning algorithm that works without knowing the transition probabilities a priori and that achieves logarithmic regret bounds, has not been developed yet for these problems.

Reinforcement learning in MDPs is considered by numerous researchers [14], [15]. In these works, it is assumed that the underlying MDP is unknown but ergodic, i.e., it is possible to reach from any state to all other states with a positive probability under any policy. These works adopt the principle of optimism under uncertainty to choose an action that maximizes the expected reward among a set of MDP models that are consistent with the estimated transition probabilities. Unlike these works, in GRBP (i) the MDP is not ergodic, and (ii) the reward is obtained only in the terminal state and not after each chosen action.

C. Multi-armed Bandits

Over the last decade many variations of the MAB problem is studied and many different learning algorithms are proposed, including Gittins index [16], upper confidence bound policies (UCB-1, UCB-2, Normalized UCB, KL-UCB) [4]–[6], greedy policies (ϵ-greedy algorithm) [4] and Thompson sampling [17] (see [8] for a comprehensive analysis of the MAB problem). The performance of a learning algorithm for a MAB problem is computed using the notion of regret. For the stochastic MAB problem [3], the regret is defined as the difference between the total (expected) reward of the learning algorithm and an *oracle* which acts optimally based on complete knowledge of the problem parameters. It is shown that the regret grows logarithmically in the number of rounds for this problem.

GRBP can be viewed as a MAB problem in which each arm corresponds to a policy. Since the set of possible deterministic policies for the GRBP is exponential in the number of states, it is infeasible to use algorithms developed for MAB problems to directly learn the optimal policy by experimenting with different policies over different rounds.

In addition, GRBP model does not fit into the combinatorial models proposed in prior works [18]. Due to these differences, existing MAB solutions cannot solve GRBP in an efficient way. Therefore, a new learning methodology that exploits the structure of the GRBP is needed.

III. Problem Formulation

A. Definition of the GRBP

In the GRBP, the system is composed of a finite set of states $\mathcal{S} := \{D, 1, \ldots, G\}$, where integer $D = 0$ denotes the *dead-end* state and G denotes the *goal* state. The set of *initial* (starting) states is denoted by $\tilde{\mathcal{S}} := \{1, \ldots, G-1\}$. The system operates in rounds ($\rho = 1, 2, \ldots$). The initial state of each round is drawn from a probability distribution $q(s), s \in \tilde{\mathcal{S}}$ over the set of initial states $\tilde{\mathcal{S}}$, such that $1 - q(1) > 0$. The current round ends and the next round starts when the learner hits state D or G. Because of this, D and G are called *terminal states*. All other states are called non-terminal states. Each round is divided into multiple time slots in which the learner takes an action in each time slot from the action set $\mathcal{A} := \{C, F\}$ with the aim of reaching to state G. Here, C denotes the continuation action and F is the terminal action. According to Fig. 1, action C moves the learner one state to the right or to the left of the current state. Action F moves the learner directly to one of the terminal states. Let s_t^ρ denote the state at the beginning of the tth time slot of round ρ and a_t^ρ denote the action taken at the tth time slot of round ρ. The state transition probabilities for action C are given by

$$\Pr(s_{t+1}^\rho = s+1 | s_t^\rho = s, a_t^\rho = C) = p^C,\ t \geq 1,\ s \in \tilde{\mathcal{S}}$$
$$\Pr(s_{t+1}^\rho = s-1 | s_t^\rho = s, a_t^\rho = C) = p^D,\ t \geq 1,\ s \in \tilde{\mathcal{S}}$$

where $0 < p^C < 1$ and $p^C + p^D = 1$. The state transition probabilities for action F are given by

$$\Pr(s_{t+1}^\rho = G | s_t^\rho = s, a_t^\rho = F) = p^F,\ t \geq 1,\ s \in \tilde{\mathcal{S}}$$
$$\Pr(s_{t+1}^\rho = D | s_t^\rho = s, a_t^\rho = F) = 1 - p^F,\ t \geq 1,\ s \in \tilde{\mathcal{S}}$$

where $0 < p^F < 1$. If the state transition probabilities are known, each round can be modeled as a MDP and an optimal policy can be found by dynamic programming [12], [19].

B. Value Functions, Rewards and the Optimal Policy

Let $\pi = (\pi_1, \pi_2, \ldots)$, where $\pi_t : \tilde{\mathcal{S}} \to \mathcal{A}$, $t \geq 1$ represent a deterministic Markov policy. π is a stationary policy if $\pi_t = \pi_{t'}$ for all t and t'. For this case we will simply use $\pi : \tilde{\mathcal{S}} \to \mathcal{A}$ to denote a stationary deterministic Markov policy. Since the time horizon is infinite within a round and the state transition probabilities are time-invariant, it is sufficient to search for the optimal policy within the set of stationary deterministic Markov policies, which is denoted by Π. Let $V^\pi(s)$ denote the probability of reaching to G by using policy π given that the system is in state s. Let $Q^\pi(s, a)$ denote the probability of reaching to G by taking action a in state s, and then continuing according to policy π. We have

$$Q^\pi(s, C) = p^C V^\pi(s+1) + p^D V^\pi(s-1),$$
$$Q^\pi(s, F) = p^F$$

for $s \in \tilde{\mathcal{S}}$. Hence, $V^\pi(s)$, $s \in \tilde{\mathcal{S}}$ can be computed by solving the following set of equations:

$$V^\pi(G) = 1,\ V^\pi(D) = 0,\ V^\pi(s) = Q^\pi(s, \pi(s)),\ \forall s \in \tilde{\mathcal{S}}$$

where $\pi(s)$ denotes the action selected by π in state s. The value of policy π is defined as

$$V^\pi := \sum_{s \in \tilde{\mathcal{S}}} q(s) V^\pi(s).$$

The optimal policy is denoted by

$$\pi^* := \arg\max_{\pi \in \Pi} V^\pi$$

and the value of the optimal policy is denoted by

$$V^* := \max_{\pi \in \Pi} V^\pi.$$

The optimal policy is characterized by Bellman optimality equations for all $s \in \tilde{S}$

$$\begin{aligned} V^*(s) &= \max\{p^F V^*(G), p^C V^*(s+1) + p^D V^*(s-1)\}, \\ &= \max\{p^F, p^C V^*(s+1) + p^D V^*(s-1)\}. \end{aligned} \quad (1)$$

As it is sufficient to search for the optimal policy within stationary deterministic Markov policies and since there are only two actions that can be taken in each $s \in \tilde{\mathcal{S}}$, the number of all such policies is 2^{G-1}. In Section IV, we will prove that the optimal policy for GRBP has a simple threshold form, which reduces the number of policies to learn from 2^{G-1} to 2.

C. Online Learning in the GRBP

As we described in the previous subsection, when the state transition probabilities are known, optimal solution and its probability of reaching to the goal can be found by solving Bellman optimality equations. When the learner does not know p^C and p^F, the optimal policy cannot be computed a priori, and hence needs to be learned. We define the learning loss of the learner, who is not aware of the optimal policy a priori, with respect to an oracle, who knows the optimal policy from the initial round, as the regret given by

$$\mathrm{Reg}(T) := TV^* - \sum_{\rho=1}^{T} V^{\hat{\pi}_\rho}$$

where $\hat{\pi}_\rho$ denotes the policy that is used by the learner in round ρ. Let $N_\pi(T)$ denote the number of times policy π is used by the learner by round T. For any policy π, let $\Delta_\pi := V^* - V^\pi$ denote the suboptimality gap of that policy. The regret can be rewritten as

$$\mathrm{Reg}(T) = \sum_{\pi \in \Pi} N_\pi(T) \Delta_\pi. \quad (2)$$

In this paper, we will design learning algorithms that minimize the growth rate of the expected regret, i.e., $\mathbb{E}[\mathrm{Reg}(T)]$. A straightforward way to do this will be to employ UCB1

algorithm [4] or its variants [6] by taking each policy as an arm. The result below state a logarithmic bound on the expected regret when UCB1 is used.

Theorem 1. *When UCB1 in [4] is used to select the policy to follow at the beginning of each round (with set of arms* Π*), we have*

$$\mathbb{E}[Reg(T)] = 8 \sum_{\pi: V^\pi < V^*} \frac{\log T}{\Delta_\pi} + \left(1 + \frac{\pi^2}{3}\right) \sum_{\pi \in \Pi} \Delta_\pi.$$

Proof. See [4]. □

As shown in Theorem 1, the expected regret of UCB1 depends linearly on the number of suboptimal policies. For GRBP, the number of policies can be very large. For instance, we have 2^{G-1} different stationary deterministic Markov policies for the defined problem. These imply that using UCB1 to learn the optimal policy is highly inefficient for the GRBP. The learning algorithm we propose in Section V exploits a result on the form of the optimal policy that will be derived in Section IV to learn the optimal policy in a fast manner. This learning algorithm calculates an estimated optimal policy using the estimated transition probabilities, and hence learns much faster than applying UCB1 naively. Moreover, it can even achieve bounded regret (instead of logarithmic regret) under some special cases.

IV. Form of the Optimal Policy

In this section, we prove that the optimal policy for GRBP has a threshold form. The value of the threshold depends only on the state transition probabilities and the number of states. First, we give the definition of a *stationary threshold* policy.

Definition 1. *π is a stationary threshold policy if there exists $\tau \in \{0, 1, \ldots, G-1\}$ such that $\pi(s) = C$ for all $s > \tau$ and $\pi(s) = F$ for all $s \leq \tau$. We use π_τ^{tr} to denote the stationary threshold policy with threshold τ. The set of stationary threshold policies is given by $\Pi^{tr} := \{\pi_\tau^{tr}\}_{\tau=\{0,1,\ldots,G-1\}}$.*

The next lemma constrains the set of policies that the optimal policy lies in.

Lemma 1. *In the GRBP it is always optimal to select action C at $s \in \tilde{\mathcal{S}} - \{1\}$.*

Proof. By (1), for $s \in \tilde{\mathcal{S}} - \{1\}$ we have

$$V^*(s) = \max\{p^F, p^C V^*(s+1) + p^D V^*(s-1)\}.$$

If $V^*(s) = p^F$, this implies that

$$p^C V^*(s+1) + p^D V^*(s-1) \leq p^F \Rightarrow$$
$$V^*(s-1) \leq \frac{p^F - p^C V^*(s+1)}{p^D}. \tag{3}$$

By definition,

$$p^F \leq V^*(s), \forall s \in \tilde{\mathcal{S}}. \tag{4}$$

Therefore,

$$\frac{p^F - p^C V^*(s+1)}{p^D} \leq \frac{p^F - p^C p^F}{p^D} = p^F$$

which in combination with (3) implies that $V^*(s-1) \leq p^F$. According to (4) we find that $V^*(s-1) = p^F$ must hold. Then, we conclude that if

$$V^*(s) = p^F \Rightarrow V^*(s-1) = p^F, \forall s \in \tilde{\mathcal{S}} - \{1\}.$$

This also implies that

$$V^*(s+1) \leq \frac{p^F - p^D V^*(s-1)}{p^C} = p^F.$$

Consequently, if $V^*(s) = p^F$ for some $s \in \tilde{\mathcal{S}} - \{1\}$, then

$$V^*(s) = p^F, \forall s \in \tilde{\mathcal{S}} - \{1\}. \tag{5}$$

By (5), if $V^*(s) = p^F$ for some $s \in \tilde{\mathcal{S}} - \{1\}$, then this implies that $V^*(G-1) = p^F$. Since $V^*(G) = 1$, we have

$$\begin{aligned} V^*(G-1) &= \max\{p^F, p^C + p^D p^F\} = p^F \\ &\Rightarrow p^F \geq p^C + p^D p^F \\ &\Rightarrow p^F(1 - p^D) \geq p^C \Rightarrow p^F \geq 1 \Rightarrow p^F = 1. \end{aligned}$$

This shows that unless $p^F = 1$, it is suboptimal to select action F in states $\tilde{\mathcal{S}} - \{1\}$ and since $p^F = 1$ is a trivial case, we disregard that. Hence, it is always optimal to select action C at $s \in \tilde{\mathcal{S}} - \{1\}$. □

The result of Lemma 1 holds independently from the set of transition probabilities (given that $p^F < 1$) and the number of states. Lemma 1 leaves out only two candidates for the optimal policy. The first candidate is the policy which selects action C at any state $s \in \tilde{\mathcal{S}}$. The second candidate selects action C in all states except state 1. Hence, the optimal policy is always in set $\{\pi_0^{tr}, \pi_1^{tr}\}$. This reduces the set of policies to consider from 2^{G-1} to 2. Let $r := p^D/p^C$ denote the *failure ratio* of action C. The next lemma gives the value functions for π_1^{tr} and π_0^{tr}.

Lemma 2. *In the GRBP we have*

(i) $V^{\pi_1^{tr}}(s) = \begin{cases} p^F + (1-p^F)\dfrac{1 - r^{s-1}}{1 - r^{G-1}}, & \text{when } r \neq 1 \\ p^F + (1-p^F)\dfrac{s-1}{G-1}, & \text{when } r = 1 \end{cases}$

(ii) $V^{\pi_0^{tr}}(s) = \begin{cases} \dfrac{1 - r^s}{1 - r^G}, & \text{when } r \neq 1 \\ \dfrac{s}{G}, & \text{when } r = 1 \end{cases}$

for $s \in \tilde{\mathcal{S}}$.

Proof. See our online appendix [20]. □

The form of the optimal policy is given in the following theorem.

Theorem 2. *In the GRBP, the optimal policy is $\pi_{\tau^*}^{tr}$, where*

$$\tau^* = \begin{cases} \text{sign}(p^F - \dfrac{1-r}{1-r^G}), & \text{when } r \neq 1 \\ \text{sign}(p^F - \dfrac{1}{G}), & \text{when } r = 1 \end{cases}$$

where $\text{sign}(x) = 1$ if x is nonnegative and 0 otherwise.

Proof. Since we have found in Lemma 1 that it is always optimal to select action C when the state is in $\{2,\dots,G-1\}$, to find the optimal policy, it is sufficient to compare the value functions of the two policies for $s=1$. When $r\neq 1$, this gives $\pi^*=\pi_1^{tr}$ if

$$\frac{1-r}{1-r^G}\leq p^F$$

and $\pi^*=\pi_0^{tr}$ otherwise.[1] Similarly, if $r=1$ and $1/G\leq p^F$, then $\pi^*=\pi_1^{tr}$. Otherwise, $\pi^*=\pi_0^{tr}$. Using these, the value of the optimal threshold is given as

$$\tau^*=\begin{cases}\text{sign}(p^F-\dfrac{1-r}{1-r^G}) & \text{if } r\neq 1\\ \text{sign}(p^F-\dfrac{1}{G}) & \text{if } r=1\end{cases}$$

which completes the proof. □

When $r\neq 1$, the term $(1-r)/(1-r^G)$ represents probability of hitting G starting from state 1 by always selecting action C. This probability is equal to $1/G$ when $r=1$. Because of this, it is optimal to take the terminal action in some cases for which $p^C>p^F$. Although the continuation action can move the system state in the direction of the goal state for some time, the long term chance of hitting the goal state by taking the continuation action can be lower than the chance of hitting the goal state by immediately taking the terminal action at state 1.

Equation of the boundary for which the optimal policy changes from π_0^{tr} to π_1^{tr} is

$$p^F=B(r):=(1-r)/(1-r^G) \tag{6}$$

when $r\neq 1$. This decision boundary is illustrated in Fig. 2 for different values of G. We call the region of transition probabilities for which π_0^{tr} is optimal as the *exploration* region, and the region for which π_1^{tr} is optimal as the *no-exploration* region. In exploration region, the optimal policy does not take action F in any round. Therefore, any learning algorithm that needs to learn how well action F performs, needs to explore action F. As the value of G increases, area of the exploration region decreases due to the fact that probability of hitting the goal state by only taking action C decreases.

V. An Online Learning Algorithm and Its Regret Analysis

In this section, we propose a learning algorithm that minimizes the regret when the state transition probabilities are unknown. The proposed algorithm forms estimates of state transition probabilities based on the history of state transitions, and then, uses these estimates together with the form of the optimal policy obtained in Section IV to calculate an estimated optimal policy at each round.

[1]When $(1-r)/(1-r^G)=p^F$ both π_1^{tr} and π_0^{tr} are optimal. For this case, we favor π_1^{tr} because it always ends the current round.

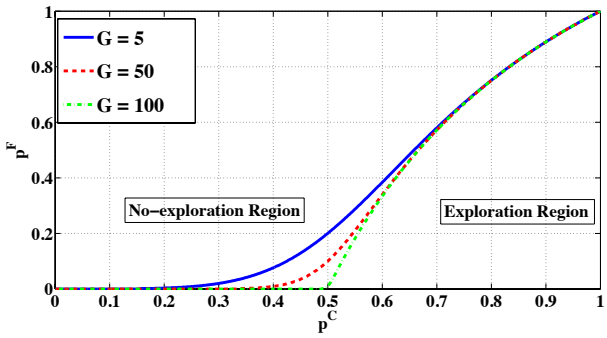

Figure 2. The boundary between exploration and no-exploration regions

A. Greedy Exploitation with Threshold Based Exploration

The learning algorithm for the GRBP is called *Greedy Exploitation with Threshold Based Exploration* (GETBE) and its pseudocode is given in Algorithm 1. Unlike conventional MAB algorithms [3], [4], [6] which require all arms to be sampled at least logarithmically many times, GETBE does not need to sample all policies (arms) logarithmically many times to find the optimal policy with a sufficiently high probability. GETBE achieves this by utilizing the form of the optimal policy derived in the previous section. Although GETBE does not require all policies to be explored, it requires exploration of action F when the estimated optimal policy never selects action F. This *forced* exploration is done to guarantee that GETBE does not get stuck in the suboptimal policy.

GETBE keeps counters $N_F^G(\rho)$, $N_F(\rho)$, $N_C^u(\rho)$ and $N_C(\rho)$: (i) $N_F^G(\rho)$ is the number of times action F is selected and terminal state G is entered upon selection of action F by the beginning of round ρ, (ii) $N_F(\rho)$ is the number of times action F is selected by the beginning of round ρ, (iii) $N_C^u(\rho)$ is the number of times transition from some state s to $s+1$ happened (i.e., the state moved up) after selecting action C by the beginning of round ρ, (iv) $N_C(\rho)$ is the number of times action C is selected by the beginning of round ρ. Let $T_F(\rho)$ and $T_C(\rho)$ represent the number of times action F and action C is selected in round ρ, respectively. Since, action F is a terminal action, it can be selected at most once in each round. However, action C can be selected multiple times in the same round. Let $T_F^G(\rho)$ and $T_C^u(\rho)$ represent the number of times state G is reached after the selection of action F and the number of times the state moved up after the selection of action C in round ρ, respectively.

At the beginning of round ρ, GETBE forms the transition probability estimates $\hat{p}_\rho^F:=N_F^G(\rho)/N_F(\rho)$ and $\hat{p}_\rho^C:=N_C^u(\rho)/N_C(\rho)$ that correspond to actions F and C, respectively. Then, it computes the estimated optimal policy $\hat{\pi}_\rho$ by using the form of the optimal policy given in Theorem 2 for the GRBP. If $\hat{\pi}_\rho=\pi_1^{tr}$, then GETBE operates in *greedy exploitation mode* by acting according to π_1^{tr} for the entire round. Else if $\hat{\pi}_\rho=\pi_0^{tr}$, then GETBE operates in *triggered exploration mode* and selects action F in the

first time slot of that round if $N_F(\rho) < D(\rho)$, where $D(\rho)$ is a non-decreasing *control function* that is an input of GETBE. This control function helps GETBE to avoid getting stuck in the suboptimal policy by forcing the selection of action F, although it is suboptimal according to $\hat{\pi}_\rho$. When $N_F(\rho) \geq D(\rho)$, GETBE employs $\hat{\pi}_\rho$ for the entire round.

At the end of round ρ the values of counters are updated as follows:

$$\begin{aligned} N_F(\rho+1) &= N_F(\rho) + T_F(\rho) \\ N_F^G(\rho+1) &= N_F^G(\rho) + T_F^G(\rho) \\ N_C(\rho+1) &= N_C(\rho) + T_C(\rho) \\ N_C^u(\rho+1) &= N_C^u(\rho) + T_C^u(\rho). \end{aligned} \tag{7}$$

These values are used to estimate the transition probabilities that will be used at the beginning of round $\rho+1$, for which the above procedure repeats. In the analysis of GETBE, we will show that when $N_F(\rho) \geq D(\rho)$, the probability that GETBE selects the suboptimal policy is very small, which implies that the regret incurred is very small.

Algorithm 1 GETBE Algorithm

Input : $G, D(\rho)$
Initialize: Take action C and then action F once to form initial estimates: $N_F^G(1), N_F(1) = 1, N_C^u(1), N_C(1) = 1$ (Round(s) to form the initial estimates (at most 2 rounds) are ignored in the regret analysis). $\rho = 1$
while $\rho \geq 1$ **do**
  Get initial state $s_1^\rho \in \tilde{\mathcal{S}}$, $t = 1$
  $\hat{p}_\rho^F = \frac{N_F^G(\rho)}{N_F(\rho)},\ \hat{p}_\rho^C = \frac{N_C^u(\rho)}{N_C(\rho)},\ \hat{r}_\rho = \frac{1-\hat{p}_\rho^C}{\hat{p}_\rho^C}$
  if $\hat{r}_\rho = 1$ **then**
    $\hat{\tau}_\rho =$ sign$(\hat{p}_\rho^F - 1/G)$
  else
    $\hat{\tau}_\rho =$ sign$(\hat{p}_\rho^F - \frac{1-\hat{r}_\rho}{1-(\hat{r}_\rho)^G})$
  end if
  Set $\hat{\pi}_\rho = \pi^{tr}_{\hat{\tau}_\rho}$
  while $s_t^\rho \neq G$ or D **do**
    if $(\hat{\pi}_\rho = \pi_0^{tr}$ && $N_F(\rho) < D(\rho))$ || $(s_t^\rho \leq \hat{\tau}_\rho)$ **then**
      Select action F, observe state s_{t+1}^ρ
      $T_F(\rho) = T_F(\rho) + 1,\ T_F^G(\rho) = \mathbb{I}(s_{t+1}^\rho = G)$[2]
    else
      Select action C, observe state s_{t+1}^ρ
      $T_C(\rho) = T_C(\rho) + 1$
      $T_C^u(\rho) = T_C^u(\rho) + \mathbb{I}(s_{t+1}^\rho = s_t^\rho + 1)$
      $t = t + 1$
    end if
  end while
  Update the counters according to (7)
  $\rho = \rho + 1$
end while

B. Regret Analysis

In this section, we bound the (expected) regret of GETBE. We show that GETBE achieves bounded regret when the unknown transition probabilities lie in no-exploration region and logarithmic (in number of rounds) regret when the unknown transition probabilities lie in exploration region. Based on Theorem 2, GETBE only needs to learn the optimal policy from the set of policies $\{\pi_0^{tr}, \pi_1^{tr}\}$. Using this fact and taking the expectation of (2), the expected regret of GETBE can be written as

$$\mathbb{E}[\text{Reg}(T)] = \sum_{\pi \in \{\pi_0^{tr}, \pi_1^{tr}\}} \mathbb{E}[N_\pi(T)]\Delta_\pi. \tag{8}$$

Let $\Delta(s) := |V^{\pi_1^{tr}}(s) - V^{\pi_0^{tr}}(s)|, s \in \tilde{\mathcal{S}}$ be the suboptimality gap when the initial state is s. For any $\pi \in \{\pi_0^{tr}, \pi_1^{tr}\}$, we have $\Delta_\pi \leq \Delta_{\max}$, where $\Delta_{\max} := \max_{s \in \tilde{\mathcal{S}}} \Delta(s)$. The next lemma gives closed-form expressions for $\Delta(s)$ and $\Delta_{\max}$.

Lemma 3. *We have*

$$\Delta(s) = \begin{cases} \frac{G-s}{G-1}|p^F - \frac{1}{G}| & \text{if } r = 1 \\ \frac{r^{G-1} - r^{s-1}}{r^{G-1}-1}|p^F - \frac{1-r}{1-r^G}| & \text{if } r \neq 1 \end{cases}$$

and

$$\Delta_{\max} = \begin{cases} |p^F - \frac{1}{G}| & \text{if } r = 1 \\ |p^F - \frac{1-r}{1-r^G}| & \text{if } r \neq 1 \end{cases}$$

Proof. See our online appendix [20]. □

Next, we bound $\mathbb{E}[N_\pi(T)]$ for the suboptimal policy in a series of lemmas. From (6), it is clear that the boundary is a function of r. Let $r = \frac{1-x}{x}$. Then, the boundary becomes a function of x by which we have

$$B(x) = (1 - \frac{1-x}{x})/(1 - (\frac{1-x}{x})^G).$$

Let δ be the minimum Euclidean distance of pair (p^C, p^F) from the boundary $(x, B(x))$ given in Fig. 2. The value of δ specifies the *hardness* of GRBP. When δ is small, it is harder to distinguish the optimal policy from the suboptimal policy. If the pair of estimated transition probabilities $(\hat{p}_\rho^C, \hat{p}_\rho^F)$ in round ρ lies within a ball around (p^C, p^F) with radius less than δ, then GETBE will select the optimal policy in that round. The probability that GETBE selects the optimal policy is lower bounded by the probability that the estimated transition probabilities lie in a ball centered at (p^C, p^F) with radius δ.

The (expected) regret given in (8) can be decomposed into two parts: (i) regret in rounds in which the suboptimal policy is selected, (ii) regret in rounds in which the optimal policy is selected and GETBE explores. Let $\text{IR}(T)$ denote the number of rounds by round T in which the suboptimal policy is selected. The first part of the regret is upper bounded by $\mathbb{E}[\text{IR}(T)]$, since the reward in a round can be either 0 or 1. Similarly, the second part of the regret is upper bounded by the number of explorations when the optimal policy is π_0^{tr}. When the optimal policy is π_1^{tr}, exploration will only be performed when the suboptimal policy is selected. Hence, there is no additional regret due to explorations, since all the regret is accounted for in the first part of the regret.

[2] $\mathbb{I}(\cdot)$ denotes the indicator function which is 1 if the expression inside evaluates true and 0 otherwise.

Let A_ρ denote the event that the suboptimal policy is selected in round ρ. Let

$$C_\rho := \{|p^C - \hat{p}^C_\rho| \geq \delta/\sqrt{2}\} \cup \{|p^F - \hat{p}^F_\rho| \geq \delta/\sqrt{2}\}.$$

It can be shown that on event C^c_ρ the Euclidian distance between (p^C, p^F) and $(\hat{p}^C_\rho, \hat{p}^F_\rho)$ is less than δ. This implies that on event C^c_ρ, the optimal policy is selected. Therefore, C_ρ contains the event that the optimal policy is not selected. Using the linearity of expectation and the union bound, we obtain

$$\begin{aligned}\mathbb{E}[\text{IR}(T)] &= \mathbb{E}[\sum_{\rho=1}^{T} \mathbb{I}(A_\rho)] \\ &\leq \sum_{\rho=1}^{T} \sum_{a \in \{F,C\}} \Pr\left(|p^{\text{a}} - \hat{p}^{\text{a}}_\rho| \geq \delta/\sqrt{2}\right). \quad (9)\end{aligned}$$

Let $\mathbb{I}^{\text{exp}}_\rho$ be the indicator function of the event that GETBE explores. By the above discussion we have

$$\mathbb{E}[\text{Reg}(T)|\pi^* = \pi^{tr}_1] \leq \mathbb{E}[\text{IR}(T)] \quad (10)$$

$$\begin{aligned}\mathbb{E}[\text{Reg}(T)|\pi^* = \pi^{tr}_0] &\leq \Delta_{\max}\mathbb{E}[\text{IR}(T)] \\ &\quad + \mathbb{E}[\sum_{\rho=1}^{T} \mathbb{I}^{\text{exp}}_\rho]. \quad (11)\end{aligned}$$

Next, we bound the expected regret of GETBE for the GRBP using (10) and (11).

Theorem 3. *When GETBE runs with $D(\rho) = \gamma \log \rho$ with $\gamma > \theta(p^F, p^C, G)$, where $\theta(p^F, p^C, G)$ is a positive constant that depends only on p^F, p^C and G, whose explicit value is given in [20], we have*

$$\mathbb{E}[\textit{Reg}(T)|\pi^* = \pi^{tr}_1] \leq w(p^F, p^C, G)$$

and

$$\mathbb{E}[\textit{Reg}(T)|\pi^* = \pi^{tr}_0] \leq \lceil D(T) \rceil + w(p^F, p^C, G)\Delta_{\max}$$

where $w(p^F, p^C, G)$ is a positive constant that depends only on p^F, p^C and G whose explicit value is given in [20].

Proof. (Sketch) For the full proof, see our online appendix [20]. First, we show that independent of the number of times π^{tr}_0 and π^{tr}_1 is selected by GETBE, $N_a(\rho)$ cannot be smaller than a logarithmic function of ρ, when ρ is sufficiently large. This is ensured for $N_C(\rho)$ since GETBE selects action C with positive probability in each round unless it explores (even when it uses π^{tr}_1). In addition, GETBE selects action F with positive probability in each round when it uses π^{tr}_1. Moreover, in rounds in which GETBE uses π^{tr}_0, it forces action F to be selected at least logarithmically many times by using the control function.

By using the above results together with Hoeffding's inequality, the bound on $\mathbb{E}[\text{IR}(T)]$ given in (9) becomes $w(p^F, p^C, G)$. In [20] it is also shown that $w(p^F, p^C, G)$ increases as δ decreases. On the other hand, $\mathbb{E}[\sum_{\rho=1}^{T} \mathbb{I}^{\text{exp}}_\rho]$ is bounded by the control function. The result follows from summing these terms using (10) and (11). □

Theorem 3 bounds the expected regret of GETBE. When $\pi^* = \pi^{tr}_1$, $\text{Reg}(T) = O(1)$ since both actions will be selected with positive probability by the optimal policy at each round. When $\pi^* = \pi^{tr}_0$, $\text{Reg}(T) = O(\log T)$ since GETBE forces to explore action F logarithmically many times to avoid getting stuck in the suboptimal policy.

VI. Numerical Results

We create a synthetic medical treatment selection problem based on [21]. Each state is assumed to be a stage of gastric cancer ($G = 4$, $D = 0$). The goal state is defined as at least three years of survival. Action C is assumed to be chemotherapy and action F is assumed to be surgery. For action C, p^C is determined by using the average survival rates for young and old groups at different stages of cancer given in [21]. For each stage, the survival rate at three years is taken to be the probability of hitting G by taking action C continuously. With this information, we set $p^C = 0.45$. Also, the five-year survival rate of surgery given in [22] (29%) is used to set $p^F = 0.3$.

The regrets shown in Fig. 3 and 4 correspond to different variants of GETBE, named as GETBE-SM, GETBE-PS and GETBE-UCB. Each variant updates the state transition probabilities in a different way. GETBE-SM uses the control function together with sample mean estimates of the state transition probabilities. Unlike GETBE-SM, GETBE-UCB and GETBE-PS do not use the control function. GETBE-PS uses posterior sampling from the Beta distribution [17] to sample and update p^F and p^C. GETBE-UCB adds an *inflation term* that is equal to $\sqrt{\frac{2\log(N_F(\rho)+N_C(\rho))}{N_a(\rho)}}$ to the sample mean estimates of the state transition probabilities that correspond to action a. PS-PolSelection and UCB-PolSelection algorithms treat each policy as a super-arm, and use PS and UCB methods to select the best policy among the two threshold policies. Instead of updating the state transition probabilities, they directly update the rewards of the policies.

Initial state distribution is taken to be the uniform distribution. Initial estimates of the transition probabilities are formed by setting $N_F(1) = 1$, $N^G_F(1) \sim \text{Unif}[0,1]$, $N_C(1) = 1$, $N^u_C(1) \sim \text{Unif}[0,1]$. The time horizon is taken to be 5000 rounds, and the control function is set to be $D(\rho) = 15 \log \rho$. Reported results are averaged over 200 iterations.

In Fig. 3 the regrets of GETBE and other algorithms are shown for p^F and p^C values given above. For this case, the the optimal policy is π^{tr}_1 and all variants of GETBE achieve finite regret, as expected. However, the regrets of UCB-PolSelection and PS-PolSelection increase logarithmically.

Next, we set $p^C = 0.65$ and $p^F = 0.3$, in order to show how the algorithms perform when the optimal policy is π^{tr}_0. The result for this case is given in Fig. 4. As expected, the regret grows logarithmically over the rounds for all variants of GETBE, PS-PolSelection and UCB-PolSelection. GETBE-PS achieves the lowest regret for this case.

Fig. 5 illustrates the regret of GETBE-SM as a function of p^F and p^C for $T = 1000$. As the state transition probabilities

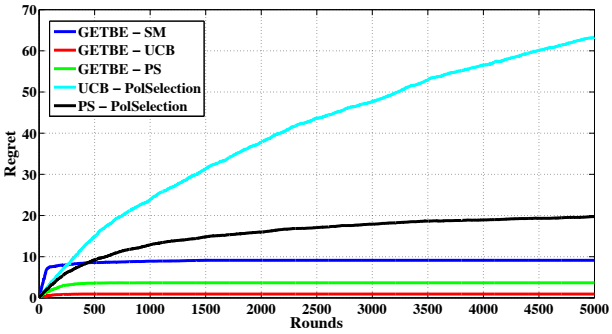

Figure 3. Regrets of GETBE and the other algorithms as a function of the number of rounds, when the transition probabilities lie in the no-exploration region.

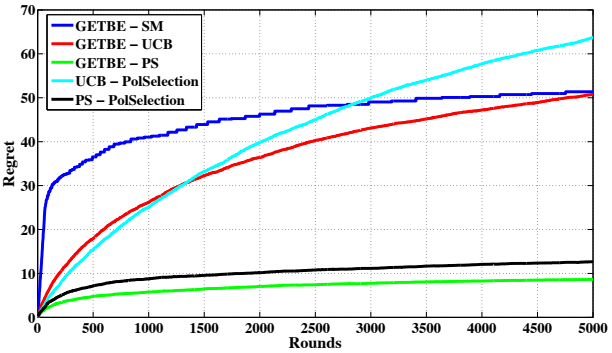

Figure 4. Regrets of GETBE and the other algorithms as a function of the number of rounds, when the transition probabilities lies in the exploration region.

shift from the no-exploration region to the exploration region the regret increases as expected.

VII. Conclusion

In this paper, we introduced the Gambler's Ruin Bandit Problem. We characterized the form of the optimal policy for this problem, and then developed a learning algorithm called GETBE that operates on the GRBP to learn the optimal policy when the transition probabilities are unknown. We proved that the regret of this algorithm is either bounded (finite) or logarithmic in the number of rounds based on the region that the true transition probabilities lie in. In addition to the regret bounds, we illustrated the performance of our algorithm via numerical experiments.

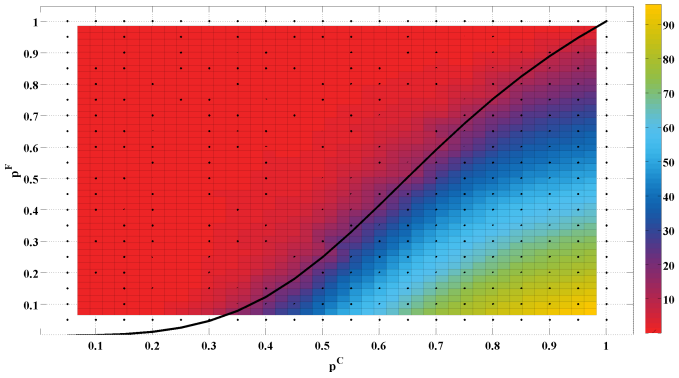

Figure 5. Regret of GETBE for different values of p^C, p^F.

References

[1] S. S. .Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges," *Statistical Science*, vol. 30, no. 2, pp. 199–215, 2015.

[2] C. Tekin and M. van der Schaar, "RELEAF: An algorithm for learning and exploiting relevance," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 716–727, 2015.

[3] Lai, T. L., Robbins, and Herbert, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[4] P. Auer, Cesa-bianchi, N. o, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.

[5] Garivier, Aurelien, Cappe, and Olivier, "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *COLT*, 2011, pp. 359–376.

[6] P. Auer and R. Ortner, "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem," *Periodica Mathematica Hungarica*, vol. 61, no. 1-2, pp. 55–65, 2010.

[7] A. Kolobov, Mausam, and D. Weld, "A theory of goal-oriented mdps with dead ends," in *UAI*, 2012, pp. 438–447.

[8] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and non-stochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[9] L. Takacs, "On the classical ruin problems," *J. Amer. Statisistical Association*, vol. 64, pp. 889–906, 1969.

[10] B. Hunter, A. C. Krinik, C. Nguyen, J. M. Switkes, and H. F. von Bremen, "Gambler's ruin with catastrophe and windfalls," *Statistical Theory and Practice*, vol. 2, no. 2, pp. 199–219, 2008.

[11] T. van Uem, "Maximum and minimum of modified gamblers ruin problem, arxiv:1301.2702," 2013.

[12] D. Bertsekas, "Dynamic programming and optimal control," *Athena Scientific*.

[13] F. Teichteil-Konigsbuch, "Stochastic safest and shortest path problems," in *AAAI*, 2012.

[14] A. Tewari and P. Bartlett, "Optimistic linear programming gives logarithmic regret for irreducible MDPs," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1505–1512, 2008.

[15] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 89–96.

[16] Gittins, J.C., Jones, and D.M., "A dynamic allocation index for the sequential design of experiments," *Progress in Statistics Gani, J. (ed.)*, pp. 241–266, 1974.

[17] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," *The Journal of Machine Learning Research*, vol. 23, no. 39, pp. 285–294, 2012.

[18] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1404–1422, 2012.

[19] R. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*. Citeseer, 1965, vol. 81.

[20] N. Akbarzadeh and C. Tekin, "Gambler's ruin bandit problem, arxiv:1605.06651," 2016.

[21] T. Isobe, K. Hashimoto, J. Kizaki, M. Miyagi, K. Aoyagi, K. Koufuji, and K. Shirouzu, "Characteristics and prognosis of gastric cancer in young patients," *International Journal of Oncology*, vol. 30, no. 1, pp. 43–49, 2013.

[22] "http://www.cancer.org/cancer/stomachcancer/detailedguide/stomach-cancer-survival-rates."